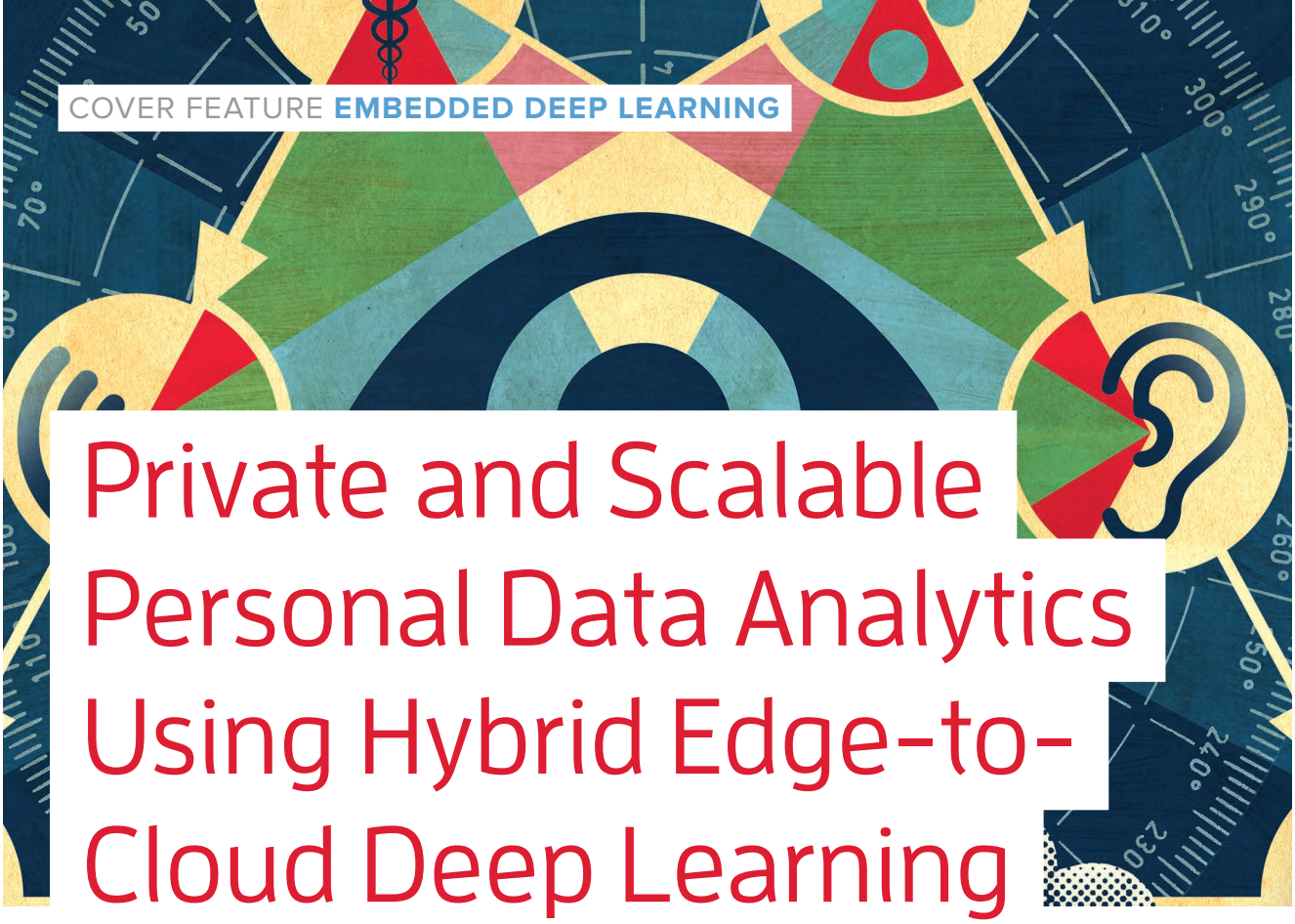

Private and Scalable Personal Data Analytics Using Hybrid Edge-to-Cloud Deep Learning

Seyed Ali Osia, Sharif University of Technology
Ali Shahin Shamsabadi, Queen Mary University of London
Ali Taheri, Sharif University of Technology
Hamid R. Rabiee, Sharif University of Technology
Hamed Haddadi, Imperial College London

Although the ability to collect, collate, and analyze the vast amount of data generated from cyber-physical systems and Internet of Things devices can be beneficial to both users and industry, this process has led to a number of challenges, including privacy and scalability issues. The authors present a hybrid framework where user-centered edge devices and resources can complement the cloud for providing privacy-aware, accurate, and efficient analytics.

The rapid rise in the development and implementation of cyber-physical systems and Internet of Things (IoT) devices is transforming our interactions with the physical world. Today, smart devices and ambient sensors are pervasively and continuously collecting and transferring large volumes of diverse user data for a variety of purposes, including security surveillance, health monitoring, and urban planning. The majority of IoT devices are constantly online by default and rely on cloud-based machine-learning applications to gain insights from the data they collect.

0018-9162/18/$33.00 © 2018 IEEE

Corporate cloud-computing services provide on-demand, high-performance, and efficient computational power and considerable cost reduction. Despite these benefits, cloud computing comes with certain challenges. Mobile and broadband bandwidth and efficiency will be a major bottleneck when the smart homes and smart cars of the next decade upload vast amounts of data from hundreds of sensors to cloud processors. These cloud-based models will also impose major energy constraints on edge devices.

Privacy issues are another important threat posed by cloud-based systems—users risk exposing their sensitive data by sharing it and allowing service providers to harvest, analyze, or monetize their data. For example, a majority of cloud-based mobile applications are free, relying on information harvesting from their users' personal data for targeted advertising. This practice has a number of privacy implications and resource impacts for users.[1] Cloud-based machine-learning algorithms can provide beneficial services (for example, health or image-based search applications), but their reliance on excessive data collection can have consequences that are unknown to the user (for example, face recognition for targeted social advertising).

Recently, edge computing has been proposed as a solution to these challenges by locating the processing power in edge nodes that are nearer to the end user—similar to fog computing at the network edge. In this way, delay-sensitive data can be analyzed on the edge nodes and cloud services can be leveraged for more delay-tolerant tasks. However, an analytics service or app provider might not be keen on sharing their valuable data-processing models. It is not always possible to assume that the feasibility of local processing (for example, a deployed deep-learning model on an edge device such as a smartphone or a computer) is a viable solution even if the task duration, memory, and processing requirements are not important for the user or if tasks can be performed when users are not actively using their devices (for example, while the device is being charged).

One could suggest that fully cryptographic–based algorithms are the ideal solution; however, the complexity of encryption methods can be high for many IoT applications, especially those relying on machine-learning models or modules that need to be continuously available or online (such as multimedia applications or sensors in a self-driving vehicle). This can be more severe for deep models, which are nonlinear, complex functions. These are difficult to estimate with polynomial functions, which are an essential component of homomorphic encryption–based methods.[2]

On one hand, complete data offloading to cloud services can have immediate or future scalability and privacy risks; on the other hand, techniques relying on performing complete analytics at the user end come with their own resource constraints (such as storage and bandwidth constraints, energy limitations, or computational costs) and user experience penalties.

In this article, we present a hybrid edge-to-cloud architecture where data processing is accomplished collaboratively between private edge data-processing units and cloud services. In this way, we can leverage edge preprocessing while addressing privacy concerns and allowing the end user to benefit from cloud-processing efficiency. A schematic view of this framework is shown in Figure 1.

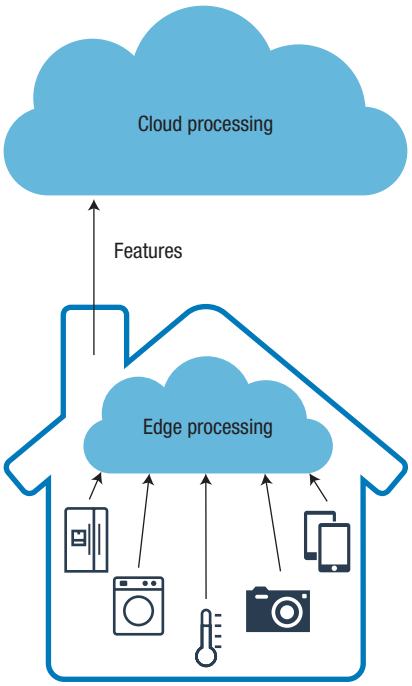

FIGURE 1. Hybrid edge-to-cloud framework for privacy-preserving machine learning. User data is collected and processed locally on private edge nodes to preserve sensitive information. The representation of data that is independent of sensitive information is sent to a cloud datacenter for applying complex inferences.

Our work focuses on achieving a compromise between resource-hungry local analytics on a private edge node and data-hungry and privacy-invasive cloud-based services. The least-necessary amount of processing takes place on the edge node, which preserves privacy, while the rest of the processing occurs in the cloud. Our main objective is to separate the feature extraction and inference phases; the former takes place locally, while the latter takes place in the cloud. With this approach, sensitive information can be removed from the data during the feature-extraction phase on the edge node, while reducing data-transmission rates to the cloud. The extracted features are transferred to the cloud server for post-processing, and the user then receives the results from the cloud.

REAL-WORLD APPLICATIONS

Advances in computer vision, machine learning, and cloud computing techniques have provided new opportunities in a large number of multimedia IoT services.[3] In this article, we explore the privacy challenges faced by these cloud-based multimedia IoT applications in the following domains.

- *Image processing*. The increasing quality of smartphone cameras and sensors, in addition to the rise in popularity of image-centric social media, has led to a variety of image analytics applications, such as scene tagging, image classification, face recognition, facial attribute prediction, age estimation, gender classification, and emotion detection.
- *Video processing*. The excessive presence of CCTV cameras shows the importance of video recording, indexing, and processing. Many homes and outdoor environments are equipped with video surveillance systems to capture visual information for different purposes. For example, smart cameras are installed in care facilities to provide patient monitoring, and autonomous vehicles use many cameras to function safely.
- *Speech processing*. Speech is increasingly becoming used in human–device interaction in the IoT domain. Many smart televisions, phones, watches, ovens, and lights have voice-command features. Increasingly, devices like Google Home and Amazon Echo are entering homes as intelligent assistants. In the next few years, speech recognition systems will become an integral part of daily life.

All of these applications require sophisticated processing of large volumes of data, usually achieved by machine-learning algorithms. Consider a classification problem such as face recognition. The classification model should be trained with a large dataset consisting of face photos labeled with the person's identity. After training, the model can label a photo with its identity. In general, machine-learning problems are supervised, unsupervised, or semi-supervised. In supervised problems, true labels are available for training data—the goal is to predict the label of test data, similar to the face-recognition example.

In this article, we focus on supervised applications, especially classification. Interested readers can refer to C.M. Bishop's *Pattern Recognition and Machine Learning*[4] to obtain more knowledge about machine learning. When true labels are not accessible, the problem is referred to as unsupervised learning or clustering. When a small number of labeled data and an abundance of unlabeled data is available, semi-supervised methods use the unlabeled data to enhance the result of supervised classification based on the labeled data.

In all of these applications, an operator might be concerned about transferring the large volume of IoT data produced at the edge of the broadband or mobile network, and clients are concerned about potential disclosure of their sensitive information. In many applications, a significant part of an individual's data does not need to be recognized by a service provider.[5] In surveillance or analytic applications, an individual's identity is the most sensitive information that is collected. For example, an individual walking by a plate-recognition camera in a parking lot should not be identifiable while classification or optical character-recognition techniques are being applied to the plate. In other words, individuals might want to be protected against undesired face-recognition models. Similarly, an individual using an IoT device voice prompt might want to be unidentifiable through their voice sessions. Privacy concerns also arise in health analytics, when application users might not want to reveal their private information.

These privacy concerns show the value and importance of a general framework that is capable of addressing privacy issues and has a long history in machine-learning applications. Training data privacy has been addressed in several previous works—for example, Charu C. Aggarwal and Philip S. Yu surveyed classic methods that consider public database privacy, such as randomization and k-anonymity.[6] In addition, much effort has been made to apply differential privacy to learning models.[7] For example, Reza Shokri and Vitaly Shmatikov[8] and Martín Abadi and his colleagues[9] attempted to make deep models differentially private. Nevertheless, less attention has been paid to user data privacy in the test phase, which is the main concern of this article.

FRAMEWORK ABSTRACTION

Let us assume we want to execute a primary task (such as speech recognition or image analysis) via cloud services. We could experience constraints due to limited local processing capabilities or conflicting commercial reasons. We also want to preserve sensitive user information (for example, the identity of a speaker could be disclosed through his voice, or an individual could be identified through CCTV

footage while walking on a public sidewalk). Hence, the data shared with the cloud service should possess two important properties: inferring the primary task is possible and deducing sensitive information is not possible.

Sharing data in the cloud provides the probability of further inferences made on sensitive information. Edge-based preprocessing of raw data can prevent revealing undesired features of the data, but such a task needs to have minimal burden due to various limitations on the client side. To achieve this, we propose a general hybrid architecture containing two main modules: a feature extractor and an analyzer. The former is constructed on a private edge node (like a personal computer or home set-top box), and the latter is stored in the cloud.

These modules and their interaction are shown in Figure 2. Data from client devices is collected on the private edge node and sent to the feature extractor, which gets the input data, applies a function to it, and outputs a set of new intermediate features, which would then be transferred to the cloud for performing the primary task. The analyzer receives the intermediate features, infers the primary information, and if needed, returns the result to the client side.

In this framework, it is critical to design a good feature-extractor module. The intermediate features need to keep the necessary information about the primary task while protecting sensitive information. As the feature extractor operates locally, it should not be a complex routine. Hence, designing this module is a challenging and important task.

As a use case, consider an image-tagging cloud service in which the identity of an individual could be exposed in an image from a live video stream. In this case, a simple feature extractor can detect faces and replace them with shaded regions. The analyzer receives this censored image and performs the image-tagging procedure (for example, it labels the image). Another common example is speech recognition, where an individual might be concerned about being identified through his or her voice. One simple solution is to simply change the pitch frequency of the voice in the feature extractor to achieve anonymity. In these two cases, designing the feature extractor is simple and will not affect the analyzer's results; however, this is not always the case.

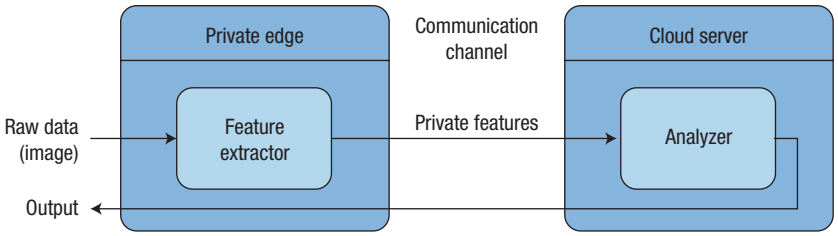

FIGURE 2. Modules of the proposed framework. The analyzer in the cloud server has access to a reduced set of private features of the data provided by the feature extractor.

In these examples, part of the data containing sensitive information is removed and the remaining data is considered the intermediate features. However, this is not applicable when the part to be removed contains important information about the primary task. For example, facial attributes like emotion or gender are also removed when removing sensitive information (the identity) by blocking a face region. Thus, we cannot use this method when our primary task is, for example, facial attribute prediction.

When the primary and sensitive information are interlocked, we encounter a complex situation. In this case, we should consider the primary task in designing the feature-extractor module for sensitive information removal. In our framework, we present a method based on deep learning, which considers both the primary task and sensitive information in the design procedure. Assuming the service provider is aware of the type of sensitive information (such as identity), the following scenario occurs: the service provider hands over a feature-extractor module to the client, which is guaranteed to consider the primary task and the sensitive information simultaneously. While the service provider does not have to share the analyzer, it must define a verification method for the privacy preservation. This process defines a privacy standard that the service providers should adopt.

DEEP-LEARNING APPLICATIONS

Deep neural networks (DNNs) have become popular in machine learning, especially in multimedia applications.[10] They provide highly accurate classifiers that extract high-level information from raw data. Deep networks consist of different layers that follow each other. Each layer is a simple function of the previous layer, representing a more sophisticated concept than its previous layers. The

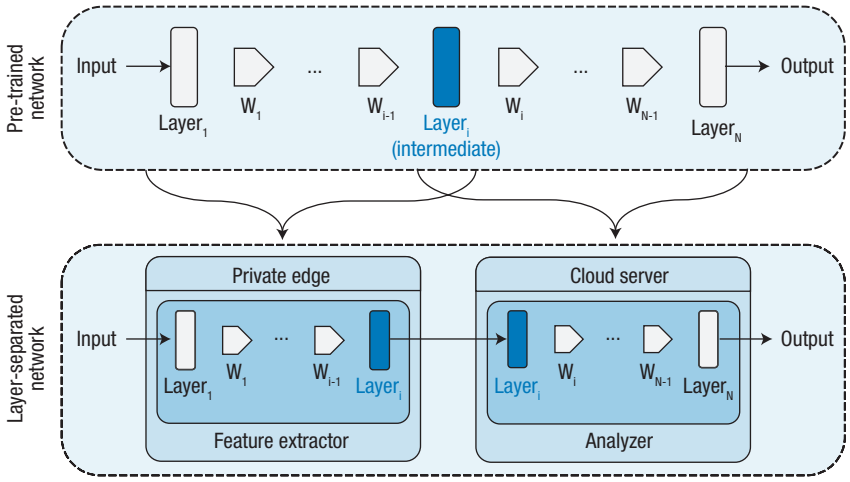

FIGURE 3. Layer-separation mechanism. Primary layers of the deep network correspond to the feature extractor, and the rest of the model is considered the analyzer.

initial layer is the raw input data and the final layer gives the inference result. All these layers together form a complex function that is applied to the input data and results in a perceptual inference. The intermediate functions are learned during the training phase via applying optimization methods on the training data. When the model is trained, it is ready to perform inference on any input data.

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are the two most well-known structures used for multimedia applications. The former is suitable for image and video processing and the latter is used mainly for sequential data processing (such as text and video). In this article, we focus on CNNs as the most popular structure for image and video processing. Suppose that inference about a primary task is done with a pre-trained deep network (such as a ready-to-use network with many layers). We address how to embed this trained model in the proposed edge-to-cloud framework as follows.

Layer separation

In deep models, the higher layers become more and more specific to the primary task, while losing other irrelevant information that contains the sensitive information we are concerned with. Based on this observation, we propose a layer-separation mechanism for a pre-trained deep network.

- First, choose an intermediate layer as a separation point.
- Then, store the layers before the intermediate layer on the edge as the feature extractor.
- Finally, store the layers after the intermediate layer in the cloud as the analyzer.

There is a tradeoff when selecting the intermediate layer—choosing it from higher layers results in higher privacy for sensitive information, but also increases the computational costs on the client side. In our previous work we provide a detailed analysis of the privacy–complexity tradeoffs for different layers, alongside the selection of the appropriate intermediate layer based on the edge device resources and user privacy constraints.[11]

We refer to this simple separation of layers between the edge and the cloud as simple embedding, as shown in Figure 3.

Siamese embedding

To increase privacy when revealing the intermediate feature to the server, we can fine-tune the existing deep model for the primary task with a particular method. Fine-tuning is a common task in training deep models. We start from a pre-trained deep model and continue its training to achieve a desired goal. As a result, we obtain an updated model that can be used in the layer-separation mechanism.

The main novelty of the proposed method relies on fine-tuning the model of the primary task by utilizing Siamese architecture[12] based on the chosen intermediate layer. Siamese architecture is a common way of training learning models, and is often used in face-verification applications to determine whether two images are of the same person. The main idea behind the Siamese network is forcing the representations of similar points (different images of the same person's face) to become near to each other, and the representations of dissimilar points (images of different people's faces) to become far from each other.

To achieve this goal, our training dataset should consist of pairs of points, which can be similar or dissimilar. For a pair of points, one function is applied to both and the distance of the two outputs is computed. Optimization is done based on a contrastive

loss function. For this loss function, the distance is maximized for two dissimilar points and minimized for two similar points. This approach makes the feature extractor more private, protecting users against inference attacks on the cloud. We refer to this as Siamese embedding.

Siamese privacy

How can we relate the Siamese architecture to privacy? Suppose our primary task is gender recognition through face portraits, accomplished by a pretrained deep model. The sensitive information is the person's identity, which should not be disclosed by using the intermediate data (for example, by a face-recognition system). In this scenario, the only thing we care about is the gender of the face portrait and not its identity. We can model this fact by defining a new similarity criterion and then fine-tuning our model with a contrastive loss function. Considering all identities with the same gender as similar not only makes the gender-recognition model more robust, but also eliminates more identity information from the intermediate features. After fine-tuning with this method, male representations are very close to each other and are far from the female representations, which are also close to each other.

Fine-tuning structure for privacy preservation is shown in Figure 4. We can apply this idea to any application by appropriately defining the similarity criterion. Experiments show that using the Siamese embedding preserves privacy while maintaining the accuracy of the primary task.

Dimensionality reduction

An important issue with all cloud-based services is their communication cost, which is usually too high. We address this concern by reducing the dimensionality of the intermediate features.

Dimensionality reduction is used in a range of applications in statistics and machine learning, from visualization to feature extraction. The dimensionality of data can be reduced by linear or nonlinear transformations of a high-dimensional space to a lower one. One of the most popular dimensionality reduction methods is the principal component analysis (PCA). PCA uses linear transformation, and the reduction and reconstruction procedures can be achieved by matrix multiplication.

The Siamese fine-tuning makes feature space much more robust in such a way that applying PCA on the fine-tuned space does not significantly decrease the accuracy of the primary task. Using dimensionality reduction on the intermediate feature space brings us two advantages without a significant reduction in primary task accuracy: it highly reduces the edge-to-cloud communication cost and it highly increases the privacy based on the nature of the reduction-reconstruction procedure.

The process of applying PCA on the intermediate feature is as follows. The service provider adds the PCA projection and reconstruction at the end of the feature extractor and the start of the analyzer, respectively. The extracted intermediate feature would be a low-dimensional vector that can be easily transferred to the cloud with low communication cost. By using these two methods, we introduce advanced embedding, in which Siamese fine-tuning is added as a pre-process and PCA projection is applied on the intermediate feature.

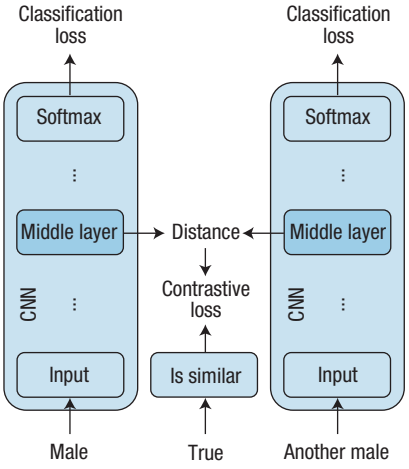

FIGURE 4. Siamese fine-tuning of the primary task. Intermediate features of two male face images are extracted via two identical convolutional neural networks (CNNs). They should be close to each other because they are considered similar.

PRELIMINARY EVALUATIONS

We performed extensive experiments on face images, with the gender classification problem as the primary task and the identity of the individual as the sensitive information to be preserved. For each of the embedding methods, we evaluated the amount of information that the intermediate feature has about gender and identity. We used an intuitive visualization technique, which demonstrates to what extent it is possible to reconstruct the original image from the intermediate data representation. We employed a more rigorous analysis of our approach in our previous work, where we proposed a privacy measure to formally quantify the ability of this framework to preserve sensitive information.[11]

To compare different deep embedding methods, we used the gender classification model proposed by Rasmus

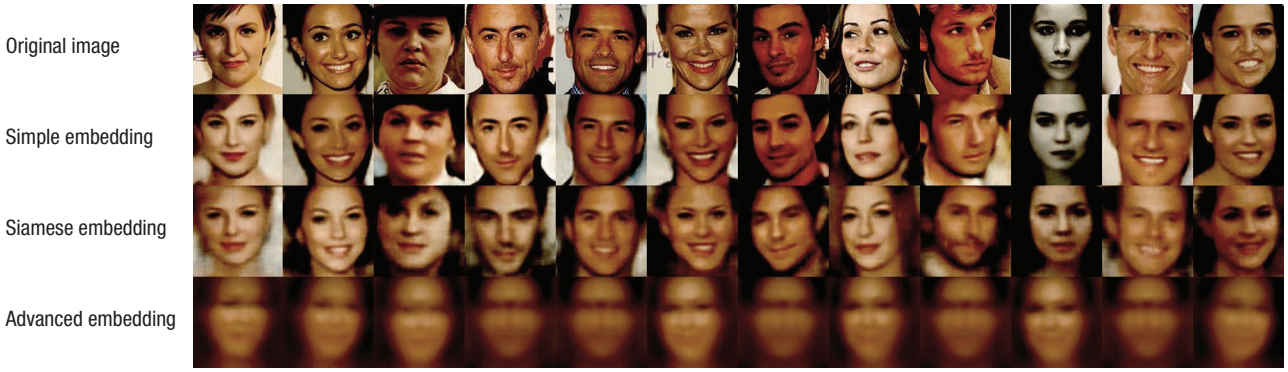

FIGURE 5. Comparison of different deep models for privacy preservation by using visualization. All methods had a similar gender classification accuracy of 93 percent. The first row shows the original images and the others show the reconstructed ones from intermediate representations. In all reconstructed images, the gender of the individuals is recognized to be the same as the originals. In addition, from simple to advanced embedding, the identity of the individuals is increasingly removed, illustrating that advanced embedding has the best privacy-preservation performance.

Rothe and his colleagues.[13] This model is a 16-layer CNN with the popular VGG-16 architecture.[14] Rothe and his colleagues collected a large dataset of faces containing age and gender attributes from IMDB and Wikipedia. Their model achieved 93 percent accuracy on the Wikipedia images. To provide a fair comparison, we also performed our experiments on this dataset.

We chose the fifth convolutional layer as our intermediate feature to build the feature-extractor module. In comparison with local device-based solutions, on average, our approach lowers the memory usage by 50 percent and the loading to less than 20 percent on a smartphone, which proves the efficiency of the proposed hybrid solution.

Simple embedding needs nothing more than layer separation. Siamese embedding is done by fine-tuning the pre-trained model and then performing the layer separation. Advanced embedding uses the same procedure with an additional process for applying PCA. We reduced the dimensions of the intermediate feature to eight. We analyzed the tradeoff between the accuracy of the gender classification (primary task) and the privacy of identity (sensitive information). Surprisingly, all these models reached almost the same accuracy of gender classification on average (93 percent). Therefore, they all had similar performances in satisfying the primary task. Hence, the only critical issue for comparison is their ability to maintain more privacy through their identity-preservation capability.

We compared these methods' privacy-preservation abilities by using a visualization technique. Visualization tries to answer a key question: Using just the intermediate layer of a deep network, what is the best recognition possibility for the original input image? Alexey Dosovitskiy and Thomas Brox answered this question by training a decoder—they used the intermediate layer as its input and the generating image as its desired output.[15] We used their method and compared the results for different deep models (although it cannot be considered rigorous proof for superior performance, it is highly intuitive).

The restored original images from intermediate features are illustrated in Figure 5 for different methods. Figure 5 shows that the genders of all images in the simple and Siamese embedding remain the same as the original images. This is also the case for the advanced embedding because of the accuracy of gender classification, although it is harder to distinguish it from the reconstructed images. The original images are almost restored in the simple embedding. Therefore, just separating layers of a deep network cannot ensure acceptable privacy-preservation performance. Siamese embedding performs better than simple embedding by distorting the identity due to intrinsic characteristics of the face (for example, the skeleton). Advanced embedding provides the best results because the decoder was not trainable and nothing can be deduced from intermediate images, including the person's identity. As an advantage of this method, the communication cost is negligible compared to other cases because we only needed to upload eight real numbers to the cloud. A more detailed analysis of this is presented in our previous work.[11]

Our framework is currently designed for pre-trained machine-learning inferences. In ongoing work, we aim to extend our method by designing a framework for machine learning as a service,[16] in which users could share their data in a privacy-preserving manner for training a new learning model in a cloud server. Another potential extension to our framework will be providing support for other kinds of neural networks such as RNNs, which are useful for temporal and sequential data processing.

REFERENCES

1. N. Vallina-Rodriguez et al., "Breaking for Commercials: Characterizing Mobile Advertising," *Proc. 2012 Internet Measurement Conference* (ICM 12), 2012, pp. 343–356.

2. R. Gilad-Bachrach et al., "Cryptonets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," *Proc. 33rd Int'l Conf. Machine Learning* (ICML 16), 2016, pp. 201–210.
3. I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Int'l J. Computer and Telecommunications Networking*, vol. 51, no. 4, 2007, pp. 921–960.
4. C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
5. A. Chaudhry et al., "Personal Data: Thinking Inside the Box," *Proc. Fifth Decennial Aarhus Conf. Critical Alternatives* (AA 15), 2015, pp. 29–32.
6. C.C. Aggarwal and P.S. Yu, "A General Survey of Privacy-Preserving Data Mining Models and Algorithms," *Privacy-Preserving Data Mining*, 2008, pp. 11–52.
7. C. Dwork, "Differential Privacy: A Survey of Results," *Int'l Conf. Theory and Applications of Models of Computation* (TAMC 08), 2008, pp. 1–19.
8. R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," *Proc. 22nd ACM SIGSAC Conf. Computer and Communications Security* (CCS 15), 2015, pp. 1310–1321.
9. M. Abadi et al., "Deep Learning with Differential Privacy," *Proc. 2016 ACM SIGSAC Conf. Computer and Communications Security* (CCS 16), 2016, pp. 308–318.
10. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, 2016.
11. S.A. Osia et al., "A Hybrid Deep Learning Architecture for Privacy-Preserving Mobile Analytics," preprint, 2017; https://arxiv.org/abs/1703.02952.
12. S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition* (CVPR 05), 2005, doi: 10.1109/CVPR.2005.202.
13. R. Rothe, R. Timofte, and L. Van Gool, "DEX: Deep EXpectation of Apparent Age from a Single Image," *2015 IEEE Int'l Conf. Computer Vision Workshop* (ICCVW 15), 2015; doi: 10.1109/ICCVW.2015.41.
14. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," preprint, 2014; https://arxiv.org/abs/1409.1556.
15. A. Dosovitskiy and T. Brox, "Inverting Visual Representations with Convolutional Networks," preprint, 2015; https://arxiv.org/abs/1506.02753.
16. S. Servia-Rodriguez et al., "Personal Model Training under Privacy Constraints," preprint, 2017; https://arxiv.org/abs/1703.00380.

ABOUT THE AUTHORS

SEYED ALI OSIA is a PhD candidate in artificial intelligence in the Department of Computer Engineering at Sharif University of Technology. His research interests include statistical machine learning, deep learning, privacy, and computer vision. Contact him at osia@ce.sharif.edu.

ALI SHAHIN SHAMSABADI is a PhD candidate in deep learning and privacy at the Centre for Intelligent Sensing, Queen Mary University of London. His research interests include deep learning and data privacy protection in distributed and centralized learning. Shamsabadi received an MSc in electrical engineering (digital) from Sharif University of Technology. Contact him at a.shahin-shamsabadi@qmul.ac.uk.

ALI TAHERI is a Master's student in the Department of Computer Engineering at Sharif University of Technology. His research interests include deep learning and privacy. Taheri received an MSc in artificial intelligence from Sharif University of Technology. Contact him at ataheri@ce.sharif.edu.

HAMID R. RABIEE is a professor of computer engineering and director of the Advanced Information and Communication Technology Research Institute (AICT), Digital Media Laboratory (DML), and Mobile Value Added Services Laboratory (MVASL) at Sharif University of Technology. He is currently on sabbatical leave as a visiting professor at Imperial College London. Rabiee's research interests include statistical machine learning, Bayesian statistics, data analytics and complex networks with applications in multimedia systems, social networks, cloud and IoT data privacy, bioinformatics, and brain networks. He received a PhD in electrical and computer engineering from Purdue University. Rabiee is a Senior Member of IEEE and holds three patents. Contact him at rabiee@sharif.edu.

HAMED HADDADI is an associate professor and the deputy director of research in the Dyson School of Design Engineering—and an academic fellow of the Data Science Institute—at Imperial College London. His research interests include user-centered systems, IoT, applied machine learning, and data security and privacy. Haddadi enjoys designing and building systems that enable better use of our digital footprint while respecting user privacy. He received a PhD in electronics engineering from University College London. Contact him at h.haddadi@imperial.ac.uk.